

# Structure-Based Bayesian Sparse Reconstruction

Ahmed A. Quadeer<sup>a</sup> and Tareq Y. Al-Naffouri<sup>b</sup>

<sup>a</sup>King Fahd University of Petroleum & Minerals, Dhahran, Saudi Arabia

<sup>b</sup>King Abdullah University of Science & Technology, Thuwal, Saudi Arabia

Email: aquadeer@kfupm.edu.sa and tareq.alnaffouri@kaust.edu.sa

## Abstract

Sparse signal reconstruction algorithms have attracted research attention due to their wide applications in various fields. In this paper, we present a simple Bayesian approach that utilizes the sparsity constraint and *a priori* statistical information (Gaussian or otherwise) to obtain near optimal estimates. In addition, we make use of the rich structure of the sensing matrix encountered in many signal processing applications to develop a fast sparse recovery algorithm. The computational complexity of the proposed algorithm is relatively low compared with the widely used convex relaxation methods as well as greedy matching pursuit techniques, especially at a low sparsity rate.<sup>1</sup>

## I. INTRODUCTION

Compressive Sensing/Compressed Sampling (CS) is a fairly new field of research that is finding many applications in statistics and signal processing [1]. As its name suggests, CS attempts to acquire a signal (inherently sparse in some subspace) at a compressed rate by randomly projecting it onto a subspace that is much smaller than the dimension of the signal itself. Provided that the sensing matrix satisfies a few conditions, the sparsity pattern of such a signal can be recovered non-combinatorially with high probability. This is in direct contrast to the traditional approach of sampling signals according to the

---

<sup>1</sup>This work was partially supported by SABIC through an internally funded project from DSR, KFUPM (Project No. SB101006) and partially by King Abdulaziz City for Science and Technology (KACST) through the Science & Technology Unit at KFUPM (Project No. 09-ELE763-04) as part of the National Science, Technology and Innovation Plan. The work of Tareq Y. Al-Naffouri was also supported by the Fullbright Scholar Program. Part of this work was presented at the Allerton Conference on Communications, Control and Computing, USA.

Nyquist theorem and then discarding the insignificant samples. Generally, most naturally occurring signals are sparse in some basis/domain and CS can therefore be utilized for their reconstruction. CS has been used successfully in, for example (but not limited to), peak-to-average power ratio reduction in orthogonal frequency division multiplexing (OFDM) [2], image processing (one-pixel camera [4]), impulse noise estimation and cancellation in power-line communication and digital subscriber lines (DSL) [5], magnetic resonance imaging (MRI) [6], channel estimation in communications systems [7], ultra-wideband (UWB) channel estimation [8], direction-of-arrival (DOA) estimation [9], and radar design [10], to name a few.

The CS problem can be set up as follows. Let  $\mathbf{x} \in \mathbb{C}^N$  be a  $P$ -sparse signal (i.e., a signal that consists of  $P$  non-zero coefficients in an  $N$ -dimensional space with  $P \ll N$ ) in some domain and let  $\mathbf{y} \in \mathbb{C}^M$  be the observation vector with  $M \ll N$  given by

$$\mathbf{y} = \Psi \mathbf{x} + \mathbf{n} \quad (1)$$

where  $\Psi$  is an  $M \times N$  measurement/sensing matrix that is assumed to be incoherent with the domain in which  $\mathbf{x}$  is sparse and  $\mathbf{n}$  is complex additive white Gaussian noise,  $\mathcal{CN}(\mathbf{0}, \sigma_n^2 \mathbf{I}_M)$ . As  $M \ll N$ , this is an ill-posed problem as there is an infinite number of solutions for  $\mathbf{x}$  satisfying (1). Now if it is known *a priori* that  $\mathbf{x}$  is sparse, the theoretical way to reconstruct the signal is to solve an  $\ell_0$ -norm minimization problem using only  $M = 2P$  measurements when the signal and measurements are free of noise [11]

$$\hat{\mathbf{x}} = \min_{\mathbf{x}} \|\mathbf{x}\|_0 \quad \text{subject to } \mathbf{y} = \Psi \mathbf{x}. \quad (2)$$

Unfortunately, solving the  $\ell_0$ -norm minimization problem is NP-hard [11] [12] and is therefore not practical. Thus, different sub-optimal approaches, categorized as compressive sensing, have been presented in the literature to solve this problem. In [12] and [13], it has been shown that  $\mathbf{x}$  can be reconstructed with high probability in polynomial time by using convex relaxation approaches at the cost of an increase in the required number of measurements. This is done by solving a relaxed  $\ell_1$ -norm minimization problem using linear programming instead of  $\ell_0$ -norm minimization [12], [13]

$$\hat{\mathbf{x}} = \min_{\mathbf{x}} \|\mathbf{x}\|_1 \quad \text{subject to } \|\mathbf{y} - \Psi \mathbf{x}\|_2 \leq \epsilon \quad (3)$$

where  $\epsilon = \sqrt{\sigma_n^2(M + \sqrt{2M})}$ . For  $\ell_1$ -norm minimization to reconstruct the sparse signal accurately, the sensing matrix,  $\Psi$ , should be sufficiently incoherent. In other words, the coherence, defined as  $\mu(\Psi) \triangleq$

$\max_{i \neq j} |\langle \psi_i, \psi_j \rangle|$ , should be as small as possible (with  $\mu(\Psi) = 1$  depicting the worst case) [12]. In [14], it has been shown that these convex relaxation approaches have a Bayesian rendition and may be viewed as maximizing the maximum *a posteriori* estimate of  $\mathbf{x}$ , given that  $\mathbf{x}$  has a Laplacian distribution. Although convex relaxation approaches are able to recover sparse signals by solving under-determined systems of equations, they also suffer from a number of drawbacks (some of which are common to other sparse recovery algorithms including [16]-[19]) that we discuss below.

#### A. Drawbacks of Convex Relaxation Approaches

1) *Complexity*: Convex relaxation relies on linear programming to solve the convex  $\ell_1$ -norm minimization problem, which is computationally relatively complex (its complexity is of the order  $\mathcal{O}(M^2 N^{3/2})$  when interior point methods are used [37]). This approach can therefore not be used in problems with very large dimensions. To overcome this drawback, many greedy algorithms have been proposed that recover the sparse signal iteratively. These include Orthogonal Matching Pursuit (OMP) [15], [16], Regularized Orthogonal Matching Pursuit (ROMP) [17], Stagewise Orthogonal Matching Pursuit (StOMP) [18], and Compressive Sampling Matching Pursuit (CoSamp) [19]. These greedy approaches are relatively faster than their convex relaxation counterparts (approximately  $\mathcal{O}(MNR)$  where  $R$  is the number of iterations).

2) *The need for randomness in the sensing matrix*: Convex relaxation methods cannot make use of the structure exhibited by the sensing matrix (e.g., a structure that comes from a Toeplitz sensing matrix or that of a partial discrete Fourier transform (DFT) matrix). In fact, if anything, this structure is harmful to these methods as the best results are obtained when the sensing matrix is close to random. This comes in contrast to current digital signal processing architectures that only deal with uniform sampling. We would thus like to employ more feasible and standard sub-sampling approaches.

3) *Inability to harness a priori statistical information*: Convex relaxation methods are not able to take account of any *a priori* statistical information (apart from sparsity information) about the signal support and additive noise. Any *a priori* statistical information can be used on the result obtained from the convex relaxation method to refine both the signal support obtained and the resulting estimate through a hypothesis testing approach [20]. However, this is only useful if these approaches are indeed able to recover the signal support. In other words, performance is bottle-necked by the support recovering capability of these approaches. We note here that the use of *a priori* statistical information for sparse signal recovery has been studied in a Bayesian context in [14] and in algorithms based on belief propagation [21], [22]. Both

[23] and [24] use *a priori* statistical information (assuming  $\mathbf{x}$  to be mixed Bernoulli-Gaussian); only [23] uses this information in a recursive manner to obtain a fast sparse signal recovery algorithm. However, it is not clear how these approaches can be extended to the non-Gaussian case.

4) *Evaluating performance in statistically familiar terms:* It is difficult to quantify the performance of convex relaxation estimates analytically in terms of the mean squared error (MSE) or bias or to relate these estimates to those obtained through more conventional approaches, e.g., maximum a posteriori probability (MAP), minimum mean-square error (MMSE), or maximum likelihood (ML).<sup>2</sup>

5) *Trading performance for computational complexity:* In general, convex relaxation approaches do not exhibit the customary tradeoff between increased computational complexity and improved recovery as is the case for, say, iterative decoding or joint channel and data detection. Rather, they solve some  $\ell_1$  problem using (second-order cone programming) with a set complexity. A number of works have attempted to derive sharp thresholds for support recovery [25], [26]. In other words, the only degree of freedom available for the designer to improve performance is to increase the number of measurements. Several iterative implementations [27], [28] of convex relaxation approaches provide some sort of flexibility by trading performance for complexity.

## B. Motivation and Paper Organization

In this paper, we present a Bayesian approach to sparse signal recovery that has low complexity and makes a collective use of 1) *a priori* statistical properties of the signal and noise, 2) sparsity information, and 3) the rich structure of the sensing matrix,  $\Psi$ . Although there have been some works that use the structure of the sensing matrix (e.g., [29]), it has not yet been rigorously exploited to aid in algorithm development and complexity reduction. We also show how our approach is able to deal with both Gaussian and non-Gaussian (or unknown) priors, and how we can compute performance measures of our estimates. In essence, we demonstrate how our technique enables us to tackle all the drawbacks of convex relaxation approaches.

This remainder of this paper is organized as follows. We start by describing the signal model in the next section. In Section III, we derive the MMSE/MAP estimates and introduce the various terms that need to be evaluated. In Section IV, we demonstrate how the structure of the sensing matrix can be

---

<sup>2</sup>It is worth noting that convex relaxation approaches have their merit in that they are agnostic to the signal distribution and thus can be quite useful when worst-case analysis is desired as opposed to average-case analysis.

used to recover the sparse signal in a divide-and-conquer manner. Section V details the proposed sparse reconstruction algorithm that we call Orthogonal Clustering. Section VI presents the different structural properties of the sensing matrix that are exploited by the proposed algorithm to reduce the computational complexity. The performance of the proposed algorithm is compared with various sparse reconstruction algorithms presented in the literature by numerical simulations in Section VII, which is followed by our conclusions in Section VIII.

### C. Notation

We denote scalars with lower-case letters (e.g.,  $x$ ), vectors with lower-case bold-faced letters (e.g.,  $\mathbf{x}$ ), matrices with upper-case, bold-faced letters (e.g.,  $\mathbf{X}$ ), and sets with script notation (e.g.  $\mathcal{S}$ ). We use  $\mathbf{x}_i$  to denote the  $i^{th}$  column of matrix  $\mathbf{X}$ ,  $x(j)$  to denote the  $j^{th}$  entry of vector  $\mathbf{x}$ , and  $\mathcal{S}_i$  to denote a subset of a set  $\mathcal{S}$ . We also use  $\mathbf{X}_{\mathcal{S}}$  to denote the sub-matrix formed by the columns  $\{\mathbf{x}_i : i \in \mathcal{S}\}$ , indexed by the set  $\mathcal{S}$ . Finally, we use  $\hat{\mathbf{x}}$ ,  $\mathbf{x}^*$ ,  $\mathbf{x}^\top$ , and  $\mathbf{x}^H$  to respectively denote the estimate, conjugate, transpose, and conjugate transpose of a vector  $\mathbf{x}$ .

## II. SIGNAL MODEL

We adopt the signal model in (1). Here, the vector  $\mathbf{x}$  is modelled as  $\mathbf{x} = \mathbf{x}_B \odot \mathbf{x}_G$ , where  $\odot$  denotes the Hadamard (element-by-element) multiplication. The entries of  $\mathbf{x}_B$  are independent and identically distributed (i.i.d) Bernoulli random variables and the entries of  $\mathbf{x}_G$  are drawn identically and independently from some zero mean distribution.<sup>3</sup> In other words, we assume that  $x_B(i)$ s are Bernoulli with success probability  $p$  and similarly that the  $x_G(i)$ s are i.i.d variables with marginal probability distribution function  $f(x)$ . The noise  $\mathbf{n}$  is assumed to be complex circularly symmetric Gaussian, i.e.,  $\mathbf{n} \sim \mathcal{CN}(0, \sigma_n^2 \mathbf{I}_M)$ . When the support set  $\mathcal{S}$  of  $\mathbf{x}$  is known, we can equivalently write (1) as

$$\mathbf{y} = \mathbf{\Psi}_{\mathcal{S}} \mathbf{x}_{\mathcal{S}} + \mathbf{n}. \quad (4)$$

## III. OPTIMUM ESTIMATION OF $\mathbf{x}$

Our task is to obtain the optimum estimate of  $\mathbf{x}$  given the observation  $\mathbf{y}$ . We can pursue either an MMSE or a MAP approach to achieve this goal. In the following, we elaborate on how we can obtain

---

<sup>3</sup>Most of the results presented in this paper also apply to the case when the entries are independent but not necessarily identically distributed.

these two estimates.

#### A. MMSE Estimation of $\mathbf{x}$

The MMSE estimate of  $\mathbf{x}$  given the observation  $\mathbf{y}$  can be expressed as

$$\hat{\mathbf{x}}_{\text{MMSE}} = \mathbb{E}[\mathbf{x}|\mathbf{y}] = \sum_{\mathcal{S}} p(\mathcal{S}|\mathbf{y}) \mathbb{E}[\mathbf{x}|\mathbf{y}, \mathcal{S}] \quad (5)$$

where the sum is over all the possible support sets  $\mathcal{S}$  of  $\mathbf{x}$ . The likelihood and expectation involved in (5) are evaluated below.

1) *Evaluation of  $\mathbb{E}[\mathbf{x}|\mathbf{y}, \mathcal{S}]$* : Recall that the relationship between  $\mathbf{y}$  and  $\mathbf{x}$  is linear (see (1)). Thus, in the case when  $\mathbf{x}$  conditioned on its support is Gaussian,  $\mathbb{E}[\mathbf{x}|\mathbf{y}, \mathcal{S}]$  is nothing but the linear MMSE estimate of  $\mathbf{x}$  given  $\mathbf{y}$  (and  $\mathcal{S}$ ), i.e.,

$$\mathbb{E}[\mathbf{x}_{\mathcal{S}}|\mathbf{y}] \triangleq \mathbb{E}[\mathbf{x}|\mathbf{y}, \mathcal{S}] = \sigma_x^2 \mathbf{\Psi}_{\mathcal{S}}^H \mathbf{\Sigma}_{\mathcal{S}}^{-1} \mathbf{y} \quad (6)$$

where

$$\mathbf{\Sigma}_{\mathcal{S}} = \frac{1}{\sigma_n^2} \mathbb{E}[\mathbf{y}\mathbf{y}^H|\mathcal{S}] = \mathbf{I}_M + \frac{\sigma_x^2}{\sigma_n^2} \mathbf{\Psi}_{\mathcal{S}} \mathbf{\Psi}_{\mathcal{S}}^H. \quad (7)$$

When  $\mathbf{x}|\mathcal{S}$  is non-Gaussian or when its statistics are unknown, the expectation  $\mathbb{E}[\mathbf{x}|\mathbf{y}, \mathcal{S}]$  is difficult or even impossible to calculate. Thus, we replace it by the best linear unbiased estimate (BLUE), i.e.,

$$\mathbb{E}[\mathbf{x}_{\mathcal{S}}|\mathbf{y}] = (\mathbf{\Psi}_{\mathcal{S}}^H \mathbf{\Psi}_{\mathcal{S}})^{-1} \mathbf{\Psi}_{\mathcal{S}}^H \mathbf{y}. \quad (8)$$

2) *Evaluation of  $p(\mathcal{S}|\mathbf{y})$* : Using Bayes' rule, we can rewrite  $p(\mathcal{S}|\mathbf{y})$  as

$$p(\mathcal{S}|\mathbf{y}) = \frac{p(\mathbf{y}|\mathcal{S})p(\mathcal{S})}{\sum_{\mathcal{S}} p(\mathbf{y}|\mathcal{S})p(\mathcal{S})}. \quad (9)$$

As the denominator  $\sum_{\mathcal{S}} p(\mathbf{y}|\mathcal{S})p(\mathcal{S})$  is common to all posterior likelihoods,  $p(\mathcal{S}|\mathbf{y})$ , it is a normalizing constant that can be ignored. To evaluate  $p(\mathcal{S})$ , note that the elements of  $\mathbf{x}$  are active according to a Bernoulli process with success probability  $p$ . Thus,  $p(\mathcal{S})$  is given by

$$p(\mathcal{S}) = p^{|\mathcal{S}|} (1-p)^{N-|\mathcal{S}|}. \quad (10)$$

It remains to evaluate  $p(\mathbf{y}|\mathcal{S})$ . Here, we distinguish between the cases of whether or not  $\mathbf{x}|\mathcal{S}$  is Gaussian.

1.  $\mathbf{x}|\mathcal{S}$  is Gaussian: When  $\mathbf{x}|\mathcal{S}$  is Gaussian,  $\mathbf{y}$  is Gaussian too with zero mean and covariance  $\mathbf{\Sigma}_{\mathcal{S}}$  and we can write the likelihood function as<sup>4</sup>

$$p(\mathbf{y}|\mathcal{S}) = \frac{\exp\left(-\frac{1}{\sigma_n^2}\|\mathbf{y}\|_{\mathbf{\Sigma}_{\mathcal{S}}^{-1}}^2\right)}{\det(\mathbf{\Sigma}_{\mathcal{S}})} \quad (11)$$

up to an irrelevant constant multiplicative factor,  $(\frac{1}{\pi^M})$ .

2.  $\mathbf{x}|\mathcal{S}$  is non-Gaussian or unknown: Alternatively, we can treat  $\mathbf{x}$  as a random vector of unknown (non-Gaussian) distribution, with support  $\mathcal{S}$ . Therefore, given the support  $\mathcal{S}$ , all we can say about  $\mathbf{y}$  is that it is formed by a vector in the subspace spanned by the columns of  $\mathbf{\Psi}_{\mathcal{S}}$ , plus a white Gaussian noise vector,  $\mathbf{n}$ . It is difficult to quantify the distribution of  $\mathbf{y}$  even if we know the distribution of (the non-Gaussian)  $\mathbf{x}$ . One way around this is to annihilate the non-Gaussian component and retain the Gaussian one. We do so by projecting  $\mathbf{y}$  onto the orthogonal complement of the span of the columns of  $\mathbf{\Psi}_{\mathcal{S}}$ , i.e., multiplying  $\mathbf{y}$  by  $\mathbf{P}_{\mathcal{S}}^{\perp} = \mathbf{I} - \mathbf{\Psi}_{\mathcal{S}}(\mathbf{\Psi}_{\mathcal{S}}^H\mathbf{\Psi}_{\mathcal{S}})^{-1}\mathbf{\Psi}_{\mathcal{S}}^H$ . This leaves us with  $\mathbf{P}_{\mathcal{S}}^{\perp}\mathbf{y} = \mathbf{P}_{\mathcal{S}}^{\perp}\mathbf{n}$ , which is zero mean and with covariance  $\mathbf{P}_{\mathcal{S}}^{\perp}\sigma_n^2\mathbf{P}_{\mathcal{S}}^{\perp H} = \sigma_n^2\mathbf{P}_{\mathcal{S}}^{\perp}$ . Thus, the conditional density of  $\mathbf{y}$  given  $\mathcal{S}$  is approximately given by

$$p(\mathbf{y}|\mathcal{S}) \simeq \exp\left(-\frac{1}{\sigma_n^2}\|\mathbf{P}_{\mathcal{S}}^{\perp}\mathbf{y}\|^2\right). \quad (12)$$

### B. MAP Estimation of $\mathbf{x}$

To obtain the MAP estimate of  $\mathbf{x}$ , we first determine the MAP estimate of  $\mathcal{S}$ , which is given by

$$\hat{\mathcal{S}}_{\text{MAP}} = \arg \max_{\mathcal{S}} p(\mathbf{y}|\mathcal{S})p(\mathcal{S}). \quad (13)$$

The prior likelihood  $p(\mathbf{y}|\mathcal{S})$ , is given by (11) when  $\mathbf{x}|\mathcal{S}$  is Gaussian and by (12) when  $\mathbf{x}|\mathcal{S}$  is non-Gaussian or unknown, whereas  $p(\mathcal{S})$  is evaluated using (10). The maximization is performed over all possible  $2^N$  support sets. The corresponding MAP estimate of  $\mathbf{x}$  is given by

$$\hat{\mathbf{x}}_{\text{MAP}} = \mathbb{E}[\mathbf{x}|\mathbf{y}, \hat{\mathcal{S}}_{\text{MAP}}]. \quad (14)$$

One can easily see that the MAP estimate is a special case of the MMSE estimate in which the sum (5) is reduced to one term. As a result, we confine the discussion in the rest of the paper to MMSE

---

<sup>4</sup> $\|\mathbf{b}\|_{\mathbf{A}}^2 \triangleq \mathbf{b}^H \mathbf{A} \mathbf{b}$

estimation.

### C. Evaluation over $\mathcal{S}$

Having evaluated the posterior probability and expectation, it remains to evaluate this over  $2^N$  possible supports (see (5) and (13)) which is a computationally daunting task. This is compounded by the fact that the calculations required for each support set in  $\mathcal{S}$  are relatively expensive, requiring some form of matrix multiplication/inversion as can be seen from (6)-(12). One way around this exhaustive approach is somehow to guess at a superset  $\mathcal{S}_r$  consisting of the most probable support and limit the sum in (5) to the superset  $\mathcal{S}_r$  and its subsets, reducing the evaluation space to  $2^{|\mathcal{S}_r|}$  points. There are two techniques that help us guess at such a set  $\mathcal{S}_r$ .

1. Convex Relaxation: Starting from (1), we can use the standard convex relaxation tools [12], [13] to find the most probable support set,  $\mathcal{S}_r$ , of the sparse vector  $\mathbf{x}$ . This is done by solving (3) and retaining some largest  $P$  non-zero values where  $P$  is selected such that  $P(\|\mathcal{S}\|_0 > P)$  is very small.<sup>5</sup>

2. Fast Bayesian Matching Pursuit (FBMP): A fast Bayesian recursive algorithm is presented in [23] that determines the dominant support and the corresponding MMSE estimate of the sparse vector.<sup>6</sup> It uses a greedy tree search over all combinations in pursuit of the dominant supports. The algorithm starts with zero active element support set. At each step, an active element is added that maximizes the Gaussian log-likelihood function similar to (11). This procedure is repeated until we reach  $P$  active elements in a branch. The procedure creates  $D$  such branches, which represent a tradeoff between performance and complexity.<sup>7</sup>

The discussion in this section applies irrespective of the type of the sensing matrix,  $\Psi$ . However, in many applications in signal processing and communications, the sensing matrix is highly structured. This fact, which has been largely overlooked in the CS literature, is utilized in the following to evaluate the MMSE (MAP) estimate at a much lower complexity than is currently available.

---

<sup>5</sup>As  $\|\mathcal{S}\|_0$  is a binomial distribution  $\sim B(N, p)$ , it can be approximated by a Gaussian distribution  $\sim \mathcal{N}(Np, Np(1-p))$ , when  $Np > 5$  (the DeMoivre-Laplace approximation [30]). In this case,  $P(\|\mathcal{S}\|_0 > P) = \frac{1}{2}\text{erfc}\left(\frac{P-N(1-p)}{\sqrt{(2Np(1-p))}}\right)$ .

<sup>6</sup>FBMP applies to the Bernoulli Gaussian case only.

<sup>7</sup>Though other greedy algorithms [16]-[19] can also be used, we focus here on FBMP as it utilizes *a priori* statistical information along with sparsity information.



#### IV. A STRUCTURE-BASED BAYESIAN RECOVERY APPROACH

Whereas in most CS literature, the sensing matrix,  $\Psi$ , is assumed to be drawn from a random constellation [12], [13], in many signal processing and communications applications, this matrix is highly structured. Thus,  $\Psi$  could be a partial DFT matrix [5] or a Toeplitz matrix (encountered in many convolution applications [7]). Table I lists various possibilities of structured  $\Psi$ .

TABLE I: Applications involving structured sensing matrices

Matrix $\Psi$	Application
Partial DFT	OFDM applications including peak-to-average power ratio reduction [2], narrow-band interference cancelation [3], and impulsive noise estimation and mitigation in DSL [5]
Toeplitz	Channel estimation [7], UWB [8], and DOA estimation [9]
Hankel	Wide-band spectrum sensing [31]
DCT	Image compression [32]
Structured Binary	Multi-user detection and contention resolution [33], [34] and feedback reduction [35], [36]

Since  $\Psi$  is a fat matrix ( $M \ll N$ ), its columns are not orthogonal (in fact not even linearly independent). However, in the aforementioned applications, one can usually find an orthogonal subset of the columns of  $\Psi$  that span the column space of  $\Psi$ . We can collect these columns into a square matrix,  $\Psi_M$ . The remaining  $N - M$  columns of  $\Psi$  group around these orthogonal columns to form semi-orthogonal clusters. In general, the columns of  $\Psi$  can be rearranged such that the farther two columns are from each other, the lower their correlation is. In this section, we demonstrate how semi-orthogonality helps to evaluate the MMSE estimate in a divide-and-conquer manner. Before we do so, we present two sensing matrices that exhibit semi-orthogonality.

##### A. Examples of Sensing Matrices with Semi-Orthogonality

1) *DFT Matrices:* We focus here on the case when the sensing matrix is a partial DFT matrix, i.e.,  $\Psi = \mathbf{S}\mathbf{F}_N$ , where  $\mathbf{F}_N$  denotes the  $N \times N$  unitary DFT matrix,  $[\mathbf{F}_N]_{a,b} = \frac{1}{\sqrt{N}}e^{-j2\pi ab/N}$  with  $a, b \in \{0, 1, \dots, N-1\}$  and  $\mathbf{S}$  is an  $M \times N$  selection matrix consisting of zeros with exactly one entry equal to 1 per row. To enforce the desired semi-orthogonal structure, the matrix  $\mathbf{S}$  usually takes the form  $\mathbf{S} = [\mathbf{O}_{M \times Z} \quad \mathbf{I}_{M \times M} \quad \mathbf{O}_{M \times (N-Z-M)}]$ , for some integer  $Z$ . In other words, the sensing matrix consists

of a continuous band of sensing frequencies. This is not unusual since in many OFDM problems, the band of interest (or the one free of transmission) is continuous. In this case, the correlation between two columns can be shown to be

$$\boldsymbol{\psi}_k^H \boldsymbol{\psi}_{k'} = \begin{cases} 1, & (k = k') \\ \left| \frac{\sin(\pi(k-k')M/N)}{M \sin(\pi(k-k')/N)} \right|, & (k \neq k') \end{cases} \quad (15)$$

which is a function of the difference,  $(k - k') \bmod N$ . It thus suffices to consider the correlation of one column with the remaining ones. Figure 1 illustrates this correlation for  $N = 1024$  and  $M = 256$ . It is worth noting that the matrix  $\boldsymbol{\Psi}$  exhibits other structural properties (e.g., the fact that it is a Vandermonde matrix), which helps us reduce the complexity of the MMSE estimation (see Section VI for further details).

2) *Toeplitz/Hankel Matrices:* We focus here on the Toeplitz case. The discussion can be easily extended to the Hankel case. A sub-sampled convolutional linear system can be written in the following matrix form,  $\mathbf{y} = \boldsymbol{\Psi}\mathbf{x} + \mathbf{n}$ , where  $\mathbf{y}$  is a vector of length  $M$ ,  $\mathbf{x}$  is a vector of length  $N$  and  $\boldsymbol{\Psi}$  is the  $M \times N$  block Toeplitz/diagonal matrix

$$\boldsymbol{\Psi} = \begin{bmatrix} \boldsymbol{\Theta} & \mathbf{O} & \cdots & \mathbf{O} \\ \mathbf{O} & \boldsymbol{\Theta} & \cdots & \mathbf{O} \\ \vdots & \ddots & \ddots & \vdots \\ \mathbf{O} & \mathbf{O} & \cdots & \boldsymbol{\Theta} \end{bmatrix}$$

where the size of  $\boldsymbol{\Theta}$  depends on the sub-sampling ratio. Here,  $\boldsymbol{\psi}_k^H \boldsymbol{\psi}_{k'} = 0$  for  $|k - k'| > L$ , and thus the columns of  $\boldsymbol{\Psi}$  can easily be grouped into truly orthogonal clusters. Note also that the individual columns of  $\boldsymbol{\Theta}$  are related to each other by a shift property, which we explore for further reduction in complexity in Section VI.

### B. Using Orthogonality for MMSE Estimation

Let  $\mathcal{S}$  be a possible support of  $\mathbf{x}$ . The columns of  $\boldsymbol{\Psi}_{\mathcal{S}}$  in (4) can be grouped into a maximum of  $C$  semi-orthogonal clusters, i.e.,  $\boldsymbol{\Psi}_{\mathcal{S}} = [\boldsymbol{\Psi}_{\mathcal{S}_1} \ \boldsymbol{\Psi}_{\mathcal{S}_2} \ \cdots \ \boldsymbol{\Psi}_{\mathcal{S}_C}]$ , where  $\mathcal{S}_i$  is the support set corresponding to

the  $i^{th}$  cluster (with  $i = 1, 2, \dots, C$ ).<sup>8</sup> Based on this fact, (4) can be written as

$$\mathbf{y} = [\Psi_{\mathcal{S}_1} \ \Psi_{\mathcal{S}_2} \ \cdots \ \Psi_{\mathcal{S}_C}] \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \\ \vdots \\ \mathbf{x}_C \end{bmatrix} + \mathbf{n}. \quad (16)$$

Columns indexed by these sets should be semi-orthogonal, i.e.,  $\Psi_{\mathcal{S}_i}^H \Psi_{\mathcal{S}_j} \simeq 0$ ; otherwise,  $\mathcal{S}_i$  and  $\mathcal{S}_j$  are merged into a bigger superset. Now, the MMSE estimate of  $\mathbf{x}$  simplifies to<sup>9</sup>

$$\hat{\mathbf{x}}_{\text{MMSE}} = \sum_{\mathcal{Z} \subset \bigcup \mathcal{S}_i} p(\mathcal{Z}|\mathbf{y}) \mathbb{E}[\mathbf{x}|\mathbf{y}, \mathcal{Z}]. \quad (17)$$

In the following, we show that  $\hat{\mathbf{x}}_{\text{MMSE}}$  can be evaluated in a divide-and-conquer manner by treating each cluster independently. To do so, we present in the following how orthogonality manifests itself in the calculation of the expectation and likelihood.

1) *The effect of orthogonality on the likelihood calculation:* Recall that up to a constant factor, the likelihood can be written as  $p(\mathcal{Z}|\mathbf{y}) = p(\mathbf{y}|\mathcal{Z})p(\mathcal{Z})$ . Now,

$$\begin{aligned} p(\mathcal{Z}) &= p\left(\bigcup \mathcal{Z}_i\right) \\ &= p^{|\bigcup \mathcal{Z}_i|} (1-p)^{N-|\bigcup \mathcal{Z}_i|} \\ &= p^{|\mathcal{Z}_1|+|\mathcal{Z}_2|+\dots+|\mathcal{Z}_C|} (1-p)^{N-(|\mathcal{Z}_1|+|\mathcal{Z}_2|+\dots+|\mathcal{Z}_C|)} \\ &= p(\mathcal{Z}_1)p(\mathcal{Z}_2)\cdots p(\mathcal{Z}_C) \end{aligned} \quad (18)$$

where the equality in (18) is true up to some constant factor. Now, to evaluate  $p(\mathbf{y}|\mathcal{Z})$ , we distinguish between the Gaussian and non-Gaussian cases. For brevity, we focus here on the Gaussian case and

---

<sup>8</sup>Here, we denote the maximum number of clusters formed by  $C$  to distinguish it from  $P$ , that refers to the estimate of the number of active supports as in [23] (see footnote 5). In our approach,  $C$  is random and depends on a threshold. This threshold is obtained using the *a priori* statistical information of the noise signal,  $\mathbf{n}$ . The procedure of forming semi-orthogonal clusters is presented in Section V.

<sup>9</sup>In writing an expression like the one in (17), it is understood that estimates of elements of  $\mathbf{x}$  that do not belong to  $\bigcup \mathcal{S}_i$  are identically zero.

extrapolate the results to the non-Gaussian case. Recall that

$$p(\mathbf{y}|\mathcal{Z}) = \frac{\exp\left(-\frac{1}{\sigma_n^2}\|\mathbf{y}\|_{\Sigma_{\mathcal{Z}}^{-1}}^2\right)}{\det(\Sigma_{\mathcal{Z}})} \quad (19)$$

with  $\Sigma_{\mathcal{Z}} = \mathbf{I}_M + \frac{\sigma_x^2}{\sigma_n^2}\Psi_{\mathcal{Z}}\Psi_{\mathcal{Z}}^H$ . Here,  $\Psi_{\mathcal{Z}} = [\Psi_{\mathcal{Z}_1} \ \Psi_{\mathcal{Z}'}]$ , where  $\Psi_{\mathcal{Z}'} = [\Psi_{\mathcal{Z}_2} \ \Psi_{\mathcal{Z}_3} \ \cdots \ \Psi_{\mathcal{Z}_C}]$ . Using the matrix inversion lemma, we can write  $\Sigma_{\mathcal{Z}}^{-1}$  as

$$\begin{aligned} \Sigma_{\mathcal{Z}}^{-1} &= (\mathbf{I}_M + \frac{\sigma_x^2}{\sigma_n^2}\Psi_{\mathcal{Z}}\Psi_{\mathcal{Z}}^H)^{-1} = (\mathbf{I}_M + \frac{\sigma_x^2}{\sigma_n^2}\Psi_{\mathcal{Z}_1}\Psi_{\mathcal{Z}_1}^H + \frac{\sigma_x^2}{\sigma_n^2}\Psi_{\mathcal{Z}'}\Psi_{\mathcal{Z}'}^H)^{-1} \\ &= \Sigma_{\mathcal{Z}_1}^{-1} - \frac{\sigma_x^2}{\sigma_n^2}\Sigma_{\mathcal{Z}_1}^{-1}\Psi_{\mathcal{Z}'}(\mathbf{I}_{\mathcal{Z}'} + \frac{\sigma_x^2}{\sigma_n^2}\Psi_{\mathcal{Z}'}^H\Sigma_{\mathcal{Z}_1}^{-1}\Psi_{\mathcal{Z}'} )^{-1}\Psi_{\mathcal{Z}'}^H\Sigma_{\mathcal{Z}_1}^{-1} \end{aligned} \quad (20)$$

where  $\Sigma_{\mathcal{Z}_1} = \mathbf{I}_M + \frac{\sigma_x^2}{\sigma_n^2}\Psi_{\mathcal{Z}_1}\Psi_{\mathcal{Z}_1}^H$ . As  $\Psi_{\mathcal{Z}_1}$  and  $\Psi_{\mathcal{Z}'}$  are almost orthogonal (i.e.,  $\Psi_{\mathcal{Z}_1}^H\Psi_{\mathcal{Z}'} = \Psi_{\mathcal{Z}'}^H\Psi_{\mathcal{Z}_1} \simeq 0$ ), (20) becomes

$$\begin{aligned} \Sigma_{\mathcal{Z}}^{-1} &= \mathbf{I}_M - \frac{\sigma_x^2}{\sigma_n^2}\Psi_{\mathcal{Z}_1}(\mathbf{I}_{\mathcal{Z}_1} + \frac{\sigma_x^2}{\sigma_n^2}\Psi_{\mathcal{Z}_1}^H\Psi_{\mathcal{Z}_1})^{-1}\Psi_{\mathcal{Z}_1}^H - \frac{\sigma_x^2}{\sigma_n^2}\Psi_{\mathcal{Z}'}(\mathbf{I}_{\mathcal{Z}'} + \frac{\sigma_x^2}{\sigma_n^2}\Psi_{\mathcal{Z}'}^H\Psi_{\mathcal{Z}'} )^{-1}\Psi_{\mathcal{Z}'}^H \\ &= -\mathbf{I}_M + \left(\mathbf{I}_M - \frac{\sigma_x^2}{\sigma_n^2}\Psi_{\mathcal{Z}_1}(\mathbf{I}_{\mathcal{Z}_1} + \frac{\sigma_x^2}{\sigma_n^2}\Psi_{\mathcal{Z}_1}^H\Psi_{\mathcal{Z}_1})^{-1}\Psi_{\mathcal{Z}_1}^H\right) + \left(\mathbf{I}_M - \frac{\sigma_x^2}{\sigma_n^2}\Psi_{\mathcal{Z}'}(\mathbf{I}_{\mathcal{Z}'} + \frac{\sigma_x^2}{\sigma_n^2}\Psi_{\mathcal{Z}'}^H\Psi_{\mathcal{Z}'} )^{-1}\Psi_{\mathcal{Z}'}^H\right) \\ &\simeq -\mathbf{I}_M + \left(\mathbf{I}_M + \frac{\sigma_x^2}{\sigma_n^2}\Psi_{\mathcal{Z}_1}\Psi_{\mathcal{Z}_1}^H\right)^{-1} + \left(\mathbf{I}_M + \frac{\sigma_x^2}{\sigma_n^2}\Psi_{\mathcal{Z}'}\Psi_{\mathcal{Z}'}^H\right)^{-1}. \end{aligned} \quad (21)$$

Continuing in the same manner, it is easy to show that

$$\Sigma_{\mathcal{Z}}^{-1} \simeq -(C-1)\mathbf{I}_M + \sum_{i=1}^C \left(\mathbf{I}_M + \frac{\sigma_x^2}{\sigma_n^2}\Psi_{\mathcal{Z}_i}\Psi_{\mathcal{Z}_i}^H\right)^{-1}. \quad (22)$$

As such, we can write

$$\exp\left(-\frac{1}{\sigma_n^2}\|\mathbf{y}\|_{\Sigma_{\mathcal{Z}}^{-1}}^2\right) \simeq \exp\left(\frac{C-1}{\sigma_n^2}\|\mathbf{y}\|^2\right) \prod_{i=1}^C \exp\left(-\frac{1}{\sigma_n^2}\|\mathbf{y}\|_{\Sigma_{\mathcal{Z}_i}^{-1}}^2\right) \quad (23)$$

where  $\Sigma_{\mathcal{Z}_i} = \mathbf{I}_M + \frac{\sigma_x^2}{\sigma_n^2}\Psi_{\mathcal{Z}_i}\Psi_{\mathcal{Z}_i}^H$ . Using a similar procedure, we can decompose  $\det(\Sigma_{\mathcal{Z}})$  as

$$\begin{aligned} \det(\Sigma_{\mathcal{Z}}) &= \det(\mathbf{I}_M + \frac{\sigma_x^2}{\sigma_n^2}\Psi_{\mathcal{Z}_1}\Psi_{\mathcal{Z}_1}^H + \frac{\sigma_x^2}{\sigma_n^2}\Psi_{\mathcal{Z}'}\Psi_{\mathcal{Z}'}^H) \\ &= \det(\mathbf{I}_M + \frac{\sigma_x^2}{\sigma_n^2}\Psi_{\mathcal{Z}_1}\Psi_{\mathcal{Z}_1}^H)\det(\mathbf{I}_M + \frac{\sigma_x^2}{\sigma_n^2}\Psi_{\mathcal{Z}'}^H\Sigma_{\mathcal{Z}_1}^{-1}\Psi_{\mathcal{Z}'} ) \end{aligned} \quad (24)$$

$$\simeq \det(\mathbf{I}_M + \frac{\sigma_x^2}{\sigma_n^2}\Psi_{\mathcal{Z}_1}\Psi_{\mathcal{Z}_1}^H)\det(\mathbf{I}_M + \frac{\sigma_x^2}{\sigma_n^2}\Psi_{\mathcal{Z}'}\Psi_{\mathcal{Z}'}^H) \quad (25)$$

$$= \det(\Sigma_{\mathcal{Z}_1})\det(\Sigma_{\mathcal{Z}'}) \quad (26)$$

where in going from (24) to (25), we used the fact that  $\Psi_{Z_1}$  and  $\Psi_{Z'}$  are almost orthogonal. Continuing in the same way, we can show that

$$\det(\Sigma_Z) \simeq \prod_{i=1}^C \det(\Sigma_{Z_i}). \quad (27)$$

Combining (23) and (27), we obtain (up to an irrelevant multiplicative factor)

$$p(\mathbf{y}|Z) \simeq \prod_{i=1}^C p(\mathbf{y}|Z_i). \quad (28)$$

Orthogonality allows us to reach the same conclusion (28) for the non-Gaussian case. Now, combining (18) and (28), we can finally write

$$p(Z|\mathbf{y}) \simeq \prod_{i=1}^C p(Z_i|\mathbf{y}) \quad (29)$$

which applies equally to the Gaussian and non-Gaussian cases.

2) *The effect of orthogonality on the expectation calculation:* In evaluating the expectation, we again distinguish between the Gaussian and non-Gaussian cases. We focus here on the non-Gaussian case for which  $\mathbb{E}[\mathbf{x}_Z|\mathbf{y}] = (\Psi_Z^H \Psi_Z)^{-1} \Psi_Z^H \mathbf{y}$ . Using the decomposition into semi-orthogonal clusters  $\Psi_Z = [\Psi_{Z_1} \ \Psi_{Z_2} \ \cdots \ \Psi_{Z_C}]$ , we can write

$$\begin{aligned} (\Psi_Z^H \Psi_Z)^{-1} \Psi_Z^H \mathbf{y} &= \begin{bmatrix} \Psi_{Z_1}^H \Psi_{Z_1} & \Psi_{Z_1}^H \Psi_{Z_2} & \cdots & \Psi_{Z_1}^H \Psi_{Z_C} \\ \vdots & \vdots & \ddots & \vdots \\ \Psi_{Z_C}^H \Psi_{Z_1} & \Psi_{Z_C}^H \Psi_{Z_2} & \cdots & \Psi_{Z_C}^H \Psi_{Z_C} \end{bmatrix}^{-1} \begin{bmatrix} \Psi_{Z_1}^H \mathbf{y} \\ \vdots \\ \Psi_{Z_C}^H \mathbf{y} \end{bmatrix} \\ &\simeq \begin{bmatrix} (\Psi_{Z_1}^H \Psi_{Z_1})^{-1} \Psi_{Z_1}^H \mathbf{y} \\ \vdots \\ (\Psi_{Z_C}^H \Psi_{Z_C})^{-1} \Psi_{Z_C}^H \mathbf{y} \end{bmatrix} \\ \text{i.e., } \mathbb{E}[\mathbf{x}_Z|\mathbf{y}] &\simeq \begin{bmatrix} \mathbb{E}[\mathbf{x}_{Z_1}|\mathbf{y}] \\ \vdots \\ \mathbb{E}[\mathbf{x}_{Z_C}|\mathbf{y}] \end{bmatrix}. \end{aligned} \quad (30)$$

Orthogonality allows us to write an identical expression to (30) in the Gaussian case.

3) *The effect of orthogonality on the MMSE estimation:* We are now ready to show how (semi)orthogonality helps with the MMSE evaluation. To do this, we substitute the decomposed expressions (29) and (30)

into (17) to get

$$\begin{aligned}
\hat{\mathbf{x}}_{\text{MMSE}} &= \sum_{\mathcal{Z} \subset \cup \mathcal{S}_i} p(\mathcal{Z}|\mathbf{y}) \mathbb{E}[\mathbf{x}|\mathbf{y}, \mathcal{Z}] \\
&\simeq \sum_{\mathcal{Z}_i \subset \mathcal{S}_i, i=1, \dots, C} \prod_i p(\mathcal{Z}_i|\mathbf{y}) \begin{bmatrix} \mathbb{E}[\mathbf{x}|\mathbf{y}, \mathcal{Z}_1] \\ \mathbb{E}[\mathbf{x}|\mathbf{y}, \mathcal{Z}_2] \\ \vdots \\ \mathbb{E}[\mathbf{x}|\mathbf{y}, \mathcal{Z}_C] \end{bmatrix} \\
&= \begin{bmatrix} \sum_{\mathcal{Z}_1 \subset \mathcal{S}_1} p(\mathcal{Z}_1|\mathbf{y}) \mathbb{E}[\mathbf{x}|\mathbf{y}, \mathcal{Z}_1] \\ \sum_{\mathcal{Z}_2 \subset \mathcal{S}_2} p(\mathcal{Z}_2|\mathbf{y}) \mathbb{E}[\mathbf{x}|\mathbf{y}, \mathcal{Z}_2] \\ \vdots \\ \sum_{\mathcal{Z}_C \subset \mathcal{S}_C} p(\mathcal{Z}_C|\mathbf{y}) \mathbb{E}[\mathbf{x}|\mathbf{y}, \mathcal{Z}_C] \end{bmatrix} \tag{31}
\end{aligned}$$

where the last line follows from the fact that  $\sum_{\mathcal{Z}_i} p(\mathcal{Z}_i|\mathbf{y}) = 1$ . Thus, the semi-orthogonality of the columns in the sensing matrix allows us to obtain the MMSE estimate of  $\mathbf{x}$  in a divide-and-conquer manner by estimating the non-overlapping sections of  $\mathbf{x}$  independently from each other. Other structural properties of  $\Psi$  can be utilized to reduce further the complexity of the MMSE estimation. For example, the orthogonal clusters exhibit some form of similarity and the columns within a particular cluster are also related to each other. We explore these properties for complexity reduction in Section VI. However, before doing so, we devote the following section to a full description of our Bayesian orthogonal clustering algorithm.

## V. AN ORTHOGONAL CLUSTERING (OC) ALGORITHM FOR SPARSE RECONSTRUCTION

In this section, we present our sparse reconstruction algorithm, which is based on orthogonal clustering. The main steps of the algorithm are detailed in the following and summarized in Figure 2.

### A. Determine dominant positions

Consider the model given in (1) reproduced here for convenience,  $\mathbf{y} = \Psi \mathbf{x} + \mathbf{n}$ . By correlating the observation vector,  $\mathbf{y}$ , with the columns of the sensing matrix,  $\Psi$ , and by retaining correlations that exceed a certain threshold, we can determine the dominant positions/regions where the support of the

sparse vector,  $\mathbf{x}$ , is located. The performance of our orthogonal clustering algorithm is dependent on this initial correlation-based guess.<sup>10</sup>

### B. Form semi-orthogonal clusters

Define a threshold  $\kappa$  such that  $p(\mathbf{n} > \kappa) \triangleq p_{\mathbf{n}}$  is very small.<sup>11</sup> The previous correlation step creates a vector of  $N$  correlations. From these correlations, obtain the indices with the correlation greater than the threshold,  $\kappa$ . Let  $i_1$  denote the index with the largest correlation above  $\kappa$  and form a cluster of size  $L$  centered around  $i_1$ .<sup>12</sup> Now, let  $i_2$  denote the corresponding index of the second largest correlation above  $\kappa$  and form another cluster of size  $L$  around  $i_2$ . If the two clusters thus formed are overlapping, merge them into one big cluster. Continue this procedure until all the correlations greater than  $\kappa$  are exhausted.

### C. Find the dominant supports and their likelihoods

Let  $L_i$  be the length of cluster  $i$  and let  $P_c$  denote the maximum possible support size in a cluster.<sup>13</sup> Let  $C$  be the total number of semi-orthogonal clusters formed in the previous step. For each of them, find the most probable support of size,  $|\mathcal{S}| = 1, |\mathcal{S}| = 2, \dots, |\mathcal{S}| = P_c$ , by calculating the likelihoods for all supports of size  $|\mathcal{S}|$  (using either (11) or (12)). Each cluster is processed independently by capitalizing on the semi-orthogonality between the clusters. The expected value of the sparse vector  $\mathbf{x}$  given  $\mathbf{y}$  and the most probable support for each size can also be evaluated using either (6) or (8) depending on the *a priori* statistical information.

### D. Evaluate the estimate of $\mathbf{x}$

Once we have the dominant supports for each cluster, their likelihoods, the expected value of  $\mathbf{x}$  given  $\mathbf{y}$  and the dominant supports, the MMSE (or MAP) estimates of  $\hat{\mathbf{x}}$  can be evaluated as discussed in

---

<sup>10</sup>We can also apply a convex relaxation approach, retain the  $P$  largest values, and form clusters around them. This allows us to incorporate *a priori* statistical information and obtain MMSE estimates but the algorithm in this case is bottle-necked by the performance of the convex relaxation approach and also loses the appeal of low complexity.

<sup>11</sup>As  $\mathbf{n} \sim \mathcal{N}(0, \sigma_n^2)$ , the threshold can be easily evaluated as,  $\kappa = \sqrt{2\sigma_n^2} \text{erfc}^{-1}(2p_{\mathbf{n}})$ .

<sup>12</sup>Given a fat sensing matrix, we consider two columns to be orthogonal (or semi orthogonal) when their correlation is below some value,  $\varepsilon$ . The cluster size  $L$  is thus the minimum separation between two columns that makes these two columns semi-orthogonal. Obviously, the distance  $L$ , is a function of the correlation tolerance,  $\varepsilon$ . The lower the tolerance,  $\varepsilon$ , the larger the cluster size,  $L$ .

<sup>13</sup> $P_c$  is calculated in a way similar to  $P$  as the support in a cluster is also a Binomial distribution  $\sim B(L_i, p)$ . Thus, we set  $P_c = \lceil \text{erfc}^{-1}(10^{-2}) \sqrt{2L_i p(1-p)} + L_i p \rceil$  (see footnote 5).

Section IV (see (31)). Note that these estimates are approximate as they are evaluated using only the dominant supports instead of using all supports.

## VI. REDUCING THE COMPUTATIONAL COMPLEXITY

In this paper, we explore three structures of the sensing matrix that help us to reduce the complexity of MMSE estimation.

- 1) *Orthogonality (independence) of clusters*: In Section IV, the orthogonality of clusters allowed us to calculate the MMSE estimate independently over clusters in a divide-and-conquer manner.
- 2) *Similarity of clusters*: While the columns of the clusters are (semi)orthogonal, allowing us to treat them independently, these columns could exhibit some form of similarity making some MMSE calculations invariant over these clusters. For example, the columns of a DFT matrix can be obtained from each other through a modulation operation while those of the Toeplitz matrix can be obtained through a shift operation. The correlation calculations that repeatedly appear in the MMSE estimation are invariant to the modulation and shift operations.
- 3) *Order within a cluster*: MMSE estimation in a cluster involves calculating the likelihoods and expectations for all supports of size  $i = 1, 2, \dots, P_c$ . Several quantities involved in these evaluations can be obtained in an order-recursive manner, incrementally moving from calculations for supports of size  $i$  to similar calculations for supports of size  $i + 1$ .

We explore the last two properties in the following subsections.

### A. Similarity of Clusters

As evident from the previous sections, calculating the likelihood can be done in a divide-and-conquer manner by calculating the likelihood for each cluster independently. This is a direct consequence of the semi-orthogonality structure of the columns of the sensing matrix. Moreover, due to the rich structure of the sensing matrix, the clusters formed are quite similar. In the following subsections, we use the structure present in DFT and Toeplitz sensing matrices to show that the likelihood and expectation expressions in each cluster (for both the Gaussian and non-Gaussian cases) are strongly related, allowing many calculations across clusters to be shared.



1) *Discrete Fourier Transform (DFT) Matrices:* Let  $\psi_1, \psi_2, \dots, \psi_L$  denote the sensing columns associated with the first cluster. Then, it is easy to see that the corresponding columns for the  $i^{th}$  cluster of equal length that are  $\Delta_i$  columns away are,  $\psi_1 \odot \psi_{\Delta_i}, \psi_2 \odot \psi_{\Delta_i}, \dots, \psi_L \odot \psi_{\Delta_i}$ , where  $\psi_{\Delta_i}$  is some constant vector that depends on the sensing columns.<sup>14</sup> Assume that we evaluate the likelihood,  $p(\mathcal{Z}_1|\mathbf{y})$ , and expectation,  $\mathbb{E}[\mathbf{x}|\mathbf{y}, \mathcal{Z}_1]$ , for a set of columns,  $\mathcal{Z}_1$ , in the first cluster. For this set, we make the assumption that

$$\mathbf{y} = \Psi_{\mathcal{Z}_1} \mathbf{x} + \mathbf{n}. \quad (32)$$

Now, let  $\mathcal{Z}_i$  denote the same set of columns chosen from the  $i^{th}$  cluster that is  $\Delta_i$  columns away (in other words  $\mathcal{Z}_i = \mathcal{Z}_1 + \Delta_i$ ). For this set, we assume that

$$\mathbf{y} = \Psi_{\mathcal{Z}_i} \mathbf{x} + \mathbf{n}. \quad (33)$$

Now (Hadamard) multiply both sides of the above equation by  $\psi_{\Delta_i}^*$  to get

$$\mathbf{y} \odot \psi_{\Delta_i}^* = \Psi_{\mathcal{Z}_1} \mathbf{x} + \mathbf{n} \odot \psi_{\Delta_i}^*. \quad (34)$$

Note that (32) and (34) have the same sensing matrix and the same noise statistics ( $\mathbf{n}$  is a white circularly symmetric Gaussian and hence is invariant to multiplication by  $\psi_{\Delta_i}^*$ ). The only difference is that  $\mathbf{y}$  is modulated by the vector  $\psi_{\Delta_i}^*$  in moving from the first to the  $i^{th}$  cluster. This allows us to write

$$p(\mathcal{Z}_i|\mathbf{y}) = p(\mathcal{Z}_1|\mathbf{y} \odot \psi_{\Delta_i}^*) \quad \text{and} \quad \mathbb{E}[\mathbf{x}|\mathbf{y}, \mathcal{Z}_i] = \mathbb{E}[\mathbf{x}|\mathbf{y} \odot \psi_{\Delta_i}^*, \mathcal{Z}_1] \quad (35)$$

which is valid for both the Gaussian and non-Gaussian cases. In other words, if  $\mathcal{Z}_i$  is obtained from  $\mathcal{Z}_1$  by a constant shift, then any  $\mathbf{y}$ -independent calculations remain the same while any calculations involving  $\mathbf{y}$  are obtained by modulating  $\mathbf{y}$  by the vector  $\psi_{\Delta_i}^*$  as shown in Figure 3. For example, the likelihood in the Gaussian case reads

$$p(\mathbf{y}|\mathcal{Z}_i) = \frac{\exp\left(-\|\mathbf{y}\|_{\Sigma_{\mathcal{Z}_i}^{-1}}^2\right)}{\det(\Sigma_{\mathcal{Z}_i})} = \frac{\exp\left(-\|\mathbf{y} \odot \psi_{\Delta_i}^*\|_{\Sigma_{\mathcal{Z}_1}^{-1}}^2\right)}{\det(\Sigma_{\mathcal{Z}_1})} \quad (36)$$

---

<sup>14</sup>For example, if we use the last  $M$  rows of the DFT matrix to construct the sensing matrix, then  $\psi_{\Delta_i} = \left[ \exp\left(-\frac{j2\pi(N-M)}{N}\Delta_i\right) \exp\left(-\frac{j2\pi(N-(M-1))}{N}\Delta_i\right) \dots \exp\left(-\frac{j2\pi(N-1)}{N}\Delta_i\right) \right]^T$ .

and, in the non-Gaussian case, it reads

$$p(\mathbf{y}|\mathcal{Z}_i) \simeq \exp\left(-\|\mathbf{y}\|_{\mathbf{P}_{\mathcal{Z}_i}^\perp}^2\right) = \exp\left(-\|\mathbf{y} \odot \boldsymbol{\psi}_{\Delta_i}^*\|_{\mathbf{P}_{\mathcal{Z}_1}^\perp}^2\right). \quad (37)$$

We observe similar behavior in calculating the expectation. Thus, in the Gaussian case, we have

$$\mathbb{E}[\mathbf{x}|\mathbf{y}, \mathcal{Z}_i] = \sigma_x^2 \boldsymbol{\Psi}_{\mathcal{Z}_i}^H \boldsymbol{\Sigma}_{\mathcal{Z}_i}^{-1} \mathbf{y} = \sigma_x^2 \boldsymbol{\Psi}_{\mathcal{Z}_1}^H \boldsymbol{\Sigma}_{\mathcal{Z}_1}^{-1} (\mathbf{y} \odot \boldsymbol{\psi}_{\Delta_i}^*) \quad (38)$$

and in the non-Gaussian case, we have

$$\mathbb{E}[\mathbf{x}|\mathbf{y}, \mathcal{Z}_i] = \left(\boldsymbol{\Psi}_{\mathcal{Z}_i}^H \boldsymbol{\Psi}_{\mathcal{Z}_i}\right)^{-1} \boldsymbol{\Psi}_{\mathcal{Z}_i}^H \mathbf{y} = \left(\boldsymbol{\Psi}_{\mathcal{Z}_1}^H \boldsymbol{\Psi}_{\mathcal{Z}_1}\right)^{-1} \boldsymbol{\Psi}_{\mathcal{Z}_1}^H (\mathbf{y} \odot \boldsymbol{\psi}_{\Delta_i}^*). \quad (39)$$

2) *Toeplitz/Hankel Matrices:* In the Toeplitz or block Toeplitz case, the sensing matrix reads  $\boldsymbol{\Psi} = [\boldsymbol{\Psi}_{\mathcal{S}_1} \ \boldsymbol{\Psi}_{\mathcal{S}_2} \ \cdots \ \boldsymbol{\Psi}_{\mathcal{S}_C}]$ . Now, the clusters can be modified to make sure that they are identical (by stretching their end points if necessary) such that  $\boldsymbol{\Psi}_{\mathcal{S}_i} = [\mathbf{O} \ \cdots \ \mathbf{O} \ \boldsymbol{\Theta}^\top \ \mathbf{O} \ \cdots \ \mathbf{O}]^\top$ . In other words, the  $\boldsymbol{\Psi}_{\mathcal{S}_i}$ s are simply shifted versions of each other. We now calculate the quantities  $\det(\boldsymbol{\Sigma}_{\mathcal{Z}_1})$ ,  $\|\mathbf{y}\|_{\boldsymbol{\Sigma}_{\mathcal{Z}_1}^{-1}}^2$ , and  $\|\mathbf{y}\|_{\mathbf{P}_{\mathcal{Z}_1}^\perp}^2$  for a set  $\mathcal{Z}_1$  of columns of the first cluster. We then choose an identical set of columns,  $\mathcal{Z}_i$ , in the  $i^{th}$  cluster. Then, it is intuitively clear that

$$\det(\boldsymbol{\Sigma}_{\mathcal{Z}_i}) = \det(\boldsymbol{\Sigma}_{\mathcal{Z}_1}), \quad \|\mathbf{y}\|_{\boldsymbol{\Sigma}_{\mathcal{Z}_i}^{-1}}^2 = \|\mathbf{y} \odot \mathbf{w}_i\|_{\boldsymbol{\Sigma}_{\mathcal{Z}_1}^{-1}}^2, \quad \text{and} \quad \|\mathbf{y}\|_{\mathbf{P}_{\mathcal{Z}_i}^\perp}^2 = \|\mathbf{y} \odot \mathbf{w}_i\|_{\mathbf{P}_{\mathcal{Z}_1}^\perp}^2 \quad (40)$$

where  $\mathbf{w}_i$  is a rectangular window corresponding to the location of the non-zero rows of  $\boldsymbol{\Psi}_{\mathcal{S}_i}$ .

### B. Order within a cluster

To evaluate the likelihood for supports of size  $i = 1, 2, \dots, P_c$  in a single cluster, we pursue an order-recursive approach, calculating the likelihood and expectation for supports of size  $i + 1$  by updating calculations made for supports of size  $i$ . In the following, we assume that we have calculated the likelihood and expectation involving the columns,  $\boldsymbol{\Psi}_{\mathcal{S}}$ , which we would like to update to  $\boldsymbol{\Psi}_{\mathcal{S}'} = [\boldsymbol{\Psi}_{\mathcal{S}} \ \boldsymbol{\psi}_i]$ .

1)  *$\mathbf{x}|S$  is Gaussian:* To calculate the likelihood  $\mathcal{L}_{\mathcal{S}'} = \frac{\exp\left(-\frac{1}{\sigma_n^2} \|\mathbf{y}\|_{\boldsymbol{\Sigma}_{\mathcal{S}'}}^2\right)}{\det(\boldsymbol{\Sigma}_{\mathcal{S}'})}$  with  $\boldsymbol{\Sigma}_{\mathcal{S}'} = \mathbf{I}_M + \frac{\sigma_x^2}{\sigma_n^2} \boldsymbol{\Psi}_{\mathcal{S}'} \boldsymbol{\Psi}_{\mathcal{S}'}^H$ , note that  $\boldsymbol{\Sigma}_{\mathcal{S}'} = \boldsymbol{\Sigma}_{\mathcal{S}} + \frac{\sigma_x^2}{\sigma_n^2} \boldsymbol{\psi}_i \boldsymbol{\psi}_i^H$ , or by the matrix inversion lemma,

$$\boldsymbol{\Sigma}_{\mathcal{S}'}^{-1} = \boldsymbol{\Sigma}_{\mathcal{S}}^{-1} - \frac{\sigma_x^2}{\sigma_n^2} \xi_i \boldsymbol{\omega}_i \boldsymbol{\omega}_i^H \quad (41)$$

where

$$\boldsymbol{\omega}_i \triangleq \boldsymbol{\Sigma}_S^{-1} \boldsymbol{\psi}_i \quad (42)$$

$$\xi_i \triangleq \left( 1 + \frac{\sigma_x^2}{\sigma_n^2} \boldsymbol{\psi}_i^H \boldsymbol{\Sigma}_S^{-1} \boldsymbol{\psi}_i \right)^{-1} = \left( 1 + \frac{\sigma_x^2}{\sigma_n^2} \boldsymbol{\psi}_i^H \boldsymbol{\omega}_i \right)^{-1}. \quad (43)$$

As we are actually interested in computing  $\exp\left(-\frac{1}{\sigma_n^2} \|\mathbf{y}\|_{\boldsymbol{\Sigma}_S^{-1}}^2\right)$ , using (41) we obtain

$$\begin{aligned} \exp\left(-\frac{1}{\sigma_n^2} \|\mathbf{y}\|_{\boldsymbol{\Sigma}_S^{-1}}^2\right) &= \exp\left(-\frac{1}{\sigma_n^2} \|\mathbf{y}\|_{\boldsymbol{\Sigma}_S^{-1}}^2 + \frac{\sigma_x^2 \xi_i}{\sigma_n^4} \|\boldsymbol{\omega}_i^H \mathbf{y}\|^2\right) \\ &= \exp\left(-\frac{1}{\sigma_n^2} \|\mathbf{y}\|_{\boldsymbol{\Sigma}_S^{-1}}^2\right) \exp\left(\frac{\sigma_x^2 \xi_i}{\sigma_n^4} \|\boldsymbol{\omega}_i^H \mathbf{y}\|^2\right). \end{aligned} \quad (44)$$

The determinant of  $\boldsymbol{\Sigma}_{S'}$  can be evaluated as follows:

$$\det(\boldsymbol{\Sigma}_{S'}) = \det\left(\boldsymbol{\Sigma}_S + \frac{\sigma_x^2}{\sigma_n^2} \boldsymbol{\psi}_i \boldsymbol{\psi}_i^H\right) = \det\left(1 + \frac{\sigma_x^2}{\sigma_n^2} \boldsymbol{\psi}_i^H \boldsymbol{\Sigma}_S^{-1} \boldsymbol{\psi}_i\right) \det(\boldsymbol{\Sigma}_S) = \xi_i^{-1} \det(\boldsymbol{\Sigma}_S). \quad (45)$$

Thus, the likelihood for the support of size  $S'$  can be written as (using (44) and (45)),

$$\begin{aligned} \mathcal{L}_{S'} &= \frac{\exp\left(-\frac{1}{\sigma_n^2} \|\mathbf{y}\|_{\boldsymbol{\Sigma}_S^{-1}}^2\right) \exp\left(\frac{\sigma_x^2 \xi_i}{\sigma_n^4} \|\boldsymbol{\omega}_i^H \mathbf{y}\|^2\right)}{\det(\boldsymbol{\Sigma}_S) \xi_i^{-1}} \\ &= \underbrace{\mathcal{L}_S \xi_i \exp\left(\frac{\sigma_x^2 \xi_i}{\sigma_n^4} \|\boldsymbol{\omega}_i^H \mathbf{y}\|^2\right)}_{\delta_i}. \end{aligned} \quad (46)$$

This shows that to calculate  $\mathcal{L}_{S'}$ , we need to compute only  $\boldsymbol{\omega}_i$  and  $\xi_i$ , which constitute  $\delta_i$ . To calculate  $\boldsymbol{\omega}_i$  for a cluster of length,  $L$ ,  $\mathcal{O}(LM^2)$  operations is required if standard matrix multiplication is used. This complexity can be reduced to  $\mathcal{O}(LM)$  by storing all the past computed values of  $\boldsymbol{\omega}$  and  $\xi$  and using the structure of  $\boldsymbol{\Sigma}_S$  [23].

Similarly,  $\mathbb{E}[\mathbf{x}_{S'} | \mathbf{y}]$  can be calculated in an order-recursive manner as follows:

$$\mathbb{E}[\mathbf{x}_{S'} | \mathbf{y}] = \begin{bmatrix} \mathbb{E}[\mathbf{x}_S | \mathbf{y}] \\ \sigma_x^2 \boldsymbol{\omega}_i^H \mathbf{y} \end{bmatrix}. \quad (47)$$

2)  $\mathbf{x}|S$  is unknown: To calculate the likelihood in the non-Gaussian case, we need to evaluate the norm,  $\|\mathbf{y}\|_{\mathbf{P}_{S'}^\perp}^2 = \|\mathbf{y}\|^2 - \mathbf{y}^H \boldsymbol{\Psi}_{S'} (\boldsymbol{\Psi}_{S'}^H \boldsymbol{\Psi}_{S'})^{-1} \boldsymbol{\Psi}_{S'}^H \mathbf{y}$ . Our approach mainly hinges on calculating the

inverse  $\mathbf{\Lambda}_{S'} \triangleq (\mathbf{\Psi}_{S'}^H \mathbf{\Psi}_{S'})^{-1}$  recursively. We do this by invoking the block inversion formula

$$\mathbf{\Lambda}_{S'} = \begin{bmatrix} \mathbf{\Lambda}_S + \frac{1}{\xi_i} \boldsymbol{\omega}_i \boldsymbol{\omega}_i^H & -\frac{1}{\xi_i} \boldsymbol{\omega}_i \\ -\frac{1}{\xi_i} \boldsymbol{\omega}_i^H & \frac{1}{\xi_i} \end{bmatrix} \quad (48)$$

$$(49)$$

where

$$\boldsymbol{\omega}_i \triangleq \mathbf{\Lambda}_S (\mathbf{\Psi}_S^H \boldsymbol{\psi}_i) \quad (50)$$

$$\xi_i \triangleq \|\boldsymbol{\psi}_i\|^2 - (\boldsymbol{\psi}_i^H \mathbf{\Psi}_S) \mathbf{\Lambda}_S (\mathbf{\Psi}_S^H \boldsymbol{\psi}_i) = \|\boldsymbol{\psi}_i\|^2 - \boldsymbol{\omega}_i^H \boldsymbol{\eta}_i \quad (51)$$

with the elements of  $\boldsymbol{\eta}_i \triangleq \mathbf{\Psi}_S^H \boldsymbol{\psi}_i$  all available (i.e., they are calculated initially and can be reused afterwards). Using this recursion, we can construct (following some straightforward manipulation) a recursion for the projected norm  $\mathcal{L}_{S'}$ :

$$\begin{aligned} \mathcal{L}_{S'} &= \exp \left( -\frac{1}{\sigma_n^2} \left[ \|\mathbf{y}\|^2 - \mathbf{y}^H \mathbf{\Psi}_{S'} \mathbf{\Lambda}_{S'} \mathbf{\Psi}_{S'}^H \mathbf{y} \right] \right) \\ &= \exp \left( -\frac{1}{\sigma_n^2} \left[ \|\mathbf{y}\|^2 - \mathbf{y}^H \mathbf{\Psi}_S \mathbf{\Lambda}_S \mathbf{\Psi}_S^H \mathbf{y} \right] \right) \\ &\quad \exp \left( -\frac{1}{\sigma_n^2} \left[ -\frac{1}{\xi_i} |(\mathbf{y}^H \mathbf{\Psi}_S) \boldsymbol{\omega}_i|^2 + \frac{2}{\xi_i} \text{Re}\{(\mathbf{y}^H \boldsymbol{\psi}_i) \boldsymbol{\omega}_i^H (\mathbf{\Psi}_S^H \mathbf{y})\} - \frac{1}{\xi_i} |\mathbf{y}^H \boldsymbol{\psi}_i|^2 \right] \right) \\ &= \underbrace{\mathcal{L}_S \exp \left( \frac{1}{\sigma_n^2 \xi_i} \left[ |(\mathbf{y}^H \mathbf{\Psi}_S) \boldsymbol{\omega}_i|^2 - 2 \text{Re}\{(\mathbf{y}^H \boldsymbol{\psi}_i) \boldsymbol{\omega}_i^H (\mathbf{\Psi}_S^H \mathbf{y})\} + |\mathbf{y}^H \boldsymbol{\psi}_i|^2 \right] \right)}_{\delta_i}. \end{aligned} \quad (52)$$

Similarly, we can show that

$$\mathbb{E}[\mathbf{x}_{S'} | \mathbf{y}] = \mathbf{\Lambda}_{S'} (\mathbf{\Psi}_{S'}^H \mathbf{y}) = \begin{bmatrix} \mathbb{E}[\mathbf{x}_S | \mathbf{y}] + \frac{1}{\xi_i} \boldsymbol{\omega}_i \boldsymbol{\eta}_i^H \mathbb{E}[\mathbf{x}_S | \mathbf{y}] - \frac{1}{\xi_i} \boldsymbol{\omega}_i \boldsymbol{\psi}_i^H \mathbf{y} \\ -\frac{1}{\xi_i} \boldsymbol{\eta}_i^H \mathbb{E}[\mathbf{x}_S | \mathbf{y}] + \frac{1}{\xi_i} \boldsymbol{\psi}_i^H \mathbf{y} \end{bmatrix}. \quad (53)$$

The cluster independent and cluster-wise evaluations in our recursive procedure for both the cases ( $\mathbf{x} | \mathcal{S}$  Gaussian or unknown) are summarized in Table II.

## VII. SIMULATION RESULTS

In this section, we compare the performance of the OC algorithm with popular sparse reconstruction methods available in the literature including the convex relaxation (CR) method [12], OMP [15], and

TABLE II: Cluster independent and cluster-wise evaluations involved in the recursive procedure for complexity reduction within a cluster

	Cluster Independent Evaluations	Cluster-wise Evaluations
$\mathbf{x} S$ is Gaussian	Evaluate $\omega_i$ and $\xi_i$ using (42) and (43) Update $\Sigma_S^{-1}$ using (41) Update $\det(\Sigma_S)$ using (45)	Evaluate $\ \omega_i^H \mathbf{y}\ ^2$ Update $\mathcal{L}_S$ using equation (46) Update $\mathbb{E}[\mathbf{x}_{S'} \mathbf{y}]$ using equation (47)
$\mathbf{x} S$ is unknown	Initialize: Calculate $\psi_i^H \psi_j \forall i, j$ Evaluate $\omega_i$ using equation (50) Update $\Sigma_S$ using equations (48) and (51)	Initialize: Evaluate $\mathbf{y}^H \psi_i \forall i$ Update $\mathcal{L}_S$ using equation (52) Update $\mathbb{E}[\mathbf{x}_{S'} \mathbf{y}]$ using equation (53)

FBMP [23]. The parameters of these algorithms are set according to the specifications provided by the authors to achieve the best results.<sup>15</sup> The parameters that we use in all the simulations are  $N = 800$ ,  $M = \frac{N}{4} = 200$ ,  $p = 10^{-2}$ , and  $\text{SNR} = 30\text{dB}$  (unless stated otherwise). Specifically, we demonstrate the performance of our algorithm for the case when the sensing matrix is a DFT or a Toeplitz matrix. We start by first investigating the effect of cluster length on the performance of OC.

#### A. The effect of the cluster length, $L$

Figure 4 compares the normalized mean-square error (NMSE) of OC as the cluster length,  $L$ , is varied. The NMSE is defined as  $\text{NMSE} = \frac{1}{R} \sum_{r=1}^R \frac{\|\hat{\mathbf{x}}^{(r)} - \mathbf{x}^{(r)}\|^2}{\|\mathbf{x}^{(r)}\|^2}$ , where  $\hat{\mathbf{x}}$  stands for the estimated sparse signal for realization  $r$ , and  $R$  is the total number of runs. For this case, the DFT matrix is used as the sensing matrix with  $\mathbf{x}|S$  Gaussian. Note that while implementing OC with fixed-length clusters, overlapping of clusters is not allowed to maintain orthogonality. This results in an increase in the probability of missing the correct support if two supports are close to each other. Thus, the smaller the cluster, the greater the probability of missing the correct supports. This is evident from Figure 4 as performance of OC improves by increasing  $L$ . Obviously, this improvement in performance is obtained at the expense of speed. Figure 5 shows that the smaller the length of clusters, the faster the algorithm. Note that for larger values of  $L$  (e.g.,  $L > 32$ ), it might not be possible to form the required number of non-overlapping clusters. To overcome this problem, we present the performance of OC implemented with variable length clusters (labeled as “OC” in Figure 4). In this case, the overlapping clusters are joined together to form larger clusters. It can be observed from Figure 4 that the performance of OC with variable-length clusters is

<sup>15</sup>For a fair comparison, we perform the MMSE refinement on the output of CR and OMP.

better than the case when it is implemented with fixed-length clusters. Moreover, this performance is achieved with a reasonable run-time<sup>16</sup> as shown in Figure 5.

### B. The effect of the signal-to-noise ratio (SNR)

Figure 6 compares the performance of the algorithms for the case when the sensing matrix is a DFT matrix and  $\mathbf{x}|\mathcal{S}$  is Gaussian. In the FBMP implementation, the number of greedy branches to explore ( $D$ ) is set to 10. Note that OC outperforms all other algorithms at low SNR while FBMP performs quite close to it at  $\text{SNR} \geq 25$  dB. It outperforms both OMP and CR at all SNR values. Specifically, at  $\text{SNR} = 25$  dB, OC has a gain of approximately 2 dB and 3 dB over CR and OMP, respectively. The performance of the algorithms for the case when the sensing matrix is a DFT matrix and  $\mathbf{x}|\mathcal{S}$  is unknown is presented in Figure 7. In this case, the entries of  $\mathbf{x}_G$  are drawn from a uniform distribution. Here, FBMP is allowed to estimate the hyper-parameters using its approximate ML algorithm (with  $E$  set to 10)[23]. It can be seen that OC easily outperforms OMP and FBMP while CR performs similar to OC. Specifically, at  $\text{SNR} = 25$  dB, OC outperforms OMP and FBMP by approximately 5 dB. Figure 8 compares the performance of the algorithms for the case when the sensing matrix is Toeplitz. To do so, we first generate a Toeplitz matrix from a column having 20 non-zero consecutive samples drawn from a Gaussian distribution. The sensing matrix is then extracted by uniformly sub-sampling this full matrix at a rate less than the duration of the signal.<sup>17</sup> Note that the performance of OC and FBMP is almost the same at low SNR but OC outperforms FBMP in the high SNR region. OMP and CR do not perform well in this case as the sensing matrix does not exhibit the requisite incoherence conditions (in this case,  $\mu(\Psi) \simeq 0.9$ ) on which much of the CS theory is based.

### C. The effect of the under-sampling ratio ( $\frac{N}{M}$ )

Figure 9 shows the performance of the algorithms (for the case when the sensing matrix is DFT and  $\mathbf{x}|\mathcal{S}$  is Gaussian) when the under-sampling ratio ( $\frac{N}{M}$ ) is varied. It can be observed that the performance of all the algorithms deteriorates as  $\frac{N}{M}$  increases. OC and FBMP perform quite close to each other with OC performing slightly better at high ( $\frac{N}{M}$ ) ratios.

---

<sup>16</sup>Thus, the following simulation results are presented with OC implemented using variable length clusters.

<sup>17</sup>In this case, the sub-sampling rate is 4 times less making  $M = 200$ .

#### D. The effect of the sparsity rate, $p$

Figure 10 compares the performance of the algorithms when the sparsity rate,  $p$ , is varied (for the case when the sensing matrix is DFT and  $\mathbf{x}|\mathcal{S}$  is Gaussian). It can be seen that the performance of OC is quite close to CR and FBMP at low sparsity rate while it outperforms OMP by approximately 3 dB for the studied range of  $p$ . The performance of OC deteriorates at the high sparsity rate because the number of clusters increases as  $p$  increases and the probability of clusters to be near or overlapping each other increases. Thus, in this case, the orthogonality assumption of OC becomes weak. Figure 11 compares the mean run-time of all the algorithms. It can be seen that OC is faster than all other algorithms. As sparsity rate increases, the length of the clusters increases, and thus the complexity of OC. Figure 12 shows that OC performs quite well at the low sparsity rate in the case when the sensing matrix is DFT and  $\mathbf{x}|\mathcal{S}$  is unknown. FBMP does not perform well at the low sparsity rate in this case even with its approximate ML algorithm. The run-time of FBMP is also higher as compared to Figure 10 due to the time taken to estimate the hyper-parameters using the ML algorithm. In the case of the Toeplitz matrix (see Figure 14), the performance of OC and FBMP is almost the same while the performance of CR and OMP is quite poor due to the weak incoherence of the sensing matrix. It can also be observed from Figure 15 that OC is quite fast compared to the other algorithms.

### VIII. CONCLUSION AND FUTURE WORK

In this paper, we present the Orthogonal Clustering algorithm for fast Bayesian sparse reconstruction. This algorithm makes collective use of the underlying structure (sparsity, *a priori* statistical information, structure of the sensing matrix) to achieve superior performance at much lower complexity compared with other algorithms especially at low sparsity rates. The proposed algorithm has the following distinctive features.

- 1) It is able to deal with Gaussian priors as well as with priors that are non-Gaussian or unknown.
- 2) It utilizes the structure of the sensing matrix, including orthogonality, modularity, and order-recursive calculations.
- 3) In the Gaussian case, OC beats all other algorithms in terms of complexity and performance for low sparsity rates. In the non-Gaussian case, it outperforms all other algorithms (most notably FBMP) for both low and high sparsity rates. Hence, the only disadvantage of OC is its performance at high

sparsity rates. In this case, the clusters are no longer orthogonal, which results in large clusters and the orthogonality assumption becomes invalid. Fortunately, this drawback is only observed in the Gaussian case while in the non-Gaussian case, OC maintains a relative advantage over the other algorithms for all sparsity rates.

- 4) It is able to provide computable measures of performance (See [5] for details on how to calculate the error covariance matrix using orthogonality).

Our future work includes

- 1) The OC algorithm assumes that various clusters do not interact. We guarantee this by lumping any two clusters that are too close into a single larger cluster. This prevents us from implementing a fixed-size cluster algorithm and gives our algorithm the advantage of being computationally cleaner and more efficient. A prerequisite to do so however is to implement an OC that takes into account the interaction between neighboring clusters.
- 2) The OC algorithm utilizes various levels of structure in the sensing matrix but falls short of utilizing one additional structure. Specifically, the various columns of any cluster are not random but are actually related (e.g., adjacent columns in the Toeplitz case exhibit a shift structure).<sup>18</sup> This additional structure can be used to reduce further the complexity of our algorithm.
- 3) The OC algorithm does not use any dependence between the active sparse elements (e.g., block sparsity). It can be specialized to deal with such cases.
- 4) The divide-and-conquer approach that we are able to pursue due to the structure of the sensing matrix can be utilized in the existing algorithms like OMP.

## REFERENCES

- [1] E. J. Candes and M. B. Wakin, "An introduction to compressive sampling," *IEEE Signal Proc. Mag.*, vol. 25, no. 2, pp. 21-30, Mar. 2008.
- [2] E. B. Al-Safadi and T. Y. Al-Naffouri, "On reducing the complexity of tone-reservation based PAPR reduction techniques by compressive sensing," *Proc. of IEEE Globecom*, Dec. 2009.
- [3] A. Gomaa and N. Al-Dhahir, "A compressive sensing approach to NBI cancellation in mobile OFDM systems," *IEEE Global Telecommunications Conf.*, pp. 1-5, USA, Dec. 2010.

---

<sup>18</sup>This structure is for example used in the lattice implementation of recursive least squares for drastic reduction in complexity [38].



- [4] M. F. Duarte, M. A. Davenport, D. Takhar, J. N. Laska, T. Sun, K. F. Kelly and R. G. Baraniuk, "Single pixel imaging via compressive sampling," *IEEE Signal Proc. Mag.*, vol. 25, no. 2, pp. 83-91, 2008.
- [5] T. Y. Al-Naffouri, A. A. Quadeer, and G. Caire, "Impulse noise estimation and cancellation in DSL using Orthogonal Clustering," *IEEE Int. Symp. Inform. Theory*, Russia, 2011.
- [6] M. Lustig, D. Donoho, and J. M. Pauly, "Sparse MRI: The application of compressed sensing for rapid MR imaging," *Magnetic Resonance in Medicine*, vol. 58, no. 6, pp. 1182-1195, 2007.
- [7] J. Haupt, W. U. Bajwa, G. Raz, and R. Nowak, "Toeplitz compressed sensing matrices with applications to sparse channel estimation," *IEEE Trans. Inform. Theory*, vol. 56, no. 11, pp. 5862-5875, Nov. 2010.
- [8] J. L. Paredes, G. R. Arce, and Z. Wang, "Ultra-Wideband compressed sensing: channel estimation," *IEEE Jnl. of Selected Topics in Signal Proc.*, vol. 1, no. 3, pp. 383-395, Oct. 2007.
- [9] Y. Wang, G. Leus, and A. Pandharipande, "Direction estimation using compressive sampling array processing," *IEEE/SP Workshop on Statistical Signal Proc.*, pp. 626-629, Cardiff, Sep. 2009.
- [10] M. Herman and T. Strohmer, "High-resolution radar via compressed sensing," *IEEE Trans. Signal Proc.*, vol. 57, no. 6, pp. 2275-2284, 2007.
- [11] R. G. Baraniuk, V. Cevher, M. F. Duarte, and C. Hegde, "Model-based compressive sensing," *IEEE Trans. Inform. Theory*, vol. 56, no. 4, pp. 1982-2001, Apr. 2010.
- [12] E. Candes, J. Romberg and T. Tao, "Stable signal recovery from incomplete and inaccurate measurements," *Comm. Pure Appl. Math*, vol. 59, no. 8, pp. 1207-1223, Aug. 2006.
- [13] E. J. Candes and P. Randall, "Highly robust error correction by convex programming," *IEEE Trans. Inform. Theory*, vol. 54, no. 7, pp. 2829-2840, Jun. 2008.
- [14] S. Ji, Y. Xue, and L. Carin, "Bayesian compressive sensing," *IEEE Trans. Signal Proc.*, vol. 56, pp. 2346-2356, 2008.
- [15] Y. C. Pati, R. Rezaifar, and P. S. Krishnaprasad, "Orthogonal matching pursuit: recursive function approximation with applications to wavelet decomposition," *Asilomar Conf. on Signals, Systems and Comput.*, Nov. 1993. (software available at <http://www.personal.soton.ac.uk/tb1m08/sparsify/sparsify.html>.)
- [16] J. Tropp and A. Gilbert, "Signal recovery from random measurements via orthogonal matching pursuit," *IEEE Trans. Inform. Theory*, vol. 53, no. 12, pp. 4655-4666, Dec. 2007.
- [17] D. Needell and R. Vershynin, "Uniform uncertainty principle and signal recovery via regularized orthogonal matching pursuit," *Foundations of Computational Mathematics*, Springer, 2009.
- [18] D. L. Donoho, Y. Tsaig, I. Drori, and J. L. Starck, "Sparse solution of underdetermined linear equations by stagewise Orthogonal Matching Pursuit (StOMP)," *Stanford technical report*.
- [19] D. Needell and J. A. Tropp, "CoSaMP: Iterative signal recovery from incomplete and inaccurate samples," *Appl. Comp. Harmonic Analysis*, 2008.
- [20] T. Y. Al-Naffouri, A. A. Quadeer, H. Hmida, and F. F. Al-Shaalan, "Impulsive noise estimation and cancellation in DSL Using compressive sensing," *IEEE Int. Symp. Circuits and Systems*, Brazil, May 2011.
- [21] X. Tan and J. Li, "Computationally efficient sparse Bayesian learning via belief propagation," *IEEE Trans. Signal Proc.*, vol. 58, no. 4, pp. 2010-2021, 2010.

- [22] A. Montanari, B. Prabhakar, and D. Tse, "Belief propagation based multi-user detection," *Allerton Conf. on Communications, Control and Computing*, USA, Sep. 2005.
- [23] P. Schniter, L. C. Potter, and J. Ziniel, "Fast Bayesian matching pursuit," *Inform. Theory and Appl. Workshop*, pp. 326-333, 2008. (software available at <http://www2.ece.ohio-state.edu/~zinielj/fbmp/download.html>.)
- [24] E. G. Larsson and Y. Selen, "Linear regression with a sparse parameter vector," *IEEE Trans. Signal Proc.*, vol. 55, no. 2, pp. 451-460, Feb. 2007.
- [25] M. Wainwright, "Information-theoretic bounds on sparsity recovery in the high-dimensional and noisy setting," *IEEE Int. Symp. Inform. Theory*, France, Jun. 2007.
- [26] A. K. Fletcher, S. Rangan, V. K. Goyal, and K. Ramchandran, "Denoising by sparse approximation: Error bounds based on rate-distortion theory," *EURASIP Jnl. Applied Signal Proc.*, vol. 2006, Article ID 26318, 19 pages, 2006.
- [27] W. Yin, S. Osher, D. Goldfarb, and J. Darbon, "Bregman iterative algorithms for  $\ell_1$ -minimization with applications to compressed sensing," *SIAM Jnl. Imaging Sciences*, vol. 1, no. 1, pp. 143-168, 2008.
- [28] E. Van den Berg and M. P. Friedlander, "SPGL1: A solver for sparse reconstruction," 2007. (software available at <http://www.cs.ubc.ca/labs/scl/spgl1>)
- [29] P. Jost, P. Vandergheynst, and P. Frossard, "Tree-based pursuit: algorithm and properties," *IEEE Trans. Signal Proc.*, vol. 54, no. 12, pp. 4685-4697, Dec. 2006.
- [30] A. Papoulis and S.U. Pillai, *Probability, random variables and stochastic processes*. 4th Edition, McGraw Hill, 2002.
- [31] Y. L. Polo, Ying Wang, A. Pandharipande, and G. Leus, "Compressive wide-band spectrum sensing," *IEEE Int. Conf. on Acoust. Speech and Signal Proc.*, pp. 2337-2340, Apr. 2009.
- [32] J. Wen, Z. Chen, Y. Han, J. D. Villasenor, and S. Yang, "A compressive sensing image compression algorithm using quantized DCT and noiselet information," *IEEE Int. Conf. on Acoust. Speech and Signal Proc.*, pp. 1294-1297, USA, Mar. 2010.
- [33] H. Zhu and G. B. Giannakis, "Sparsity-embracing multiuser detection for CDMA systems with low activity factory," *IEEE Int. Symp. Inform. Theory*, pp. 164-168, Korea, Jun. 2009.
- [34] S. T. Qaseem, T. Y. Al-Naffouri, and T. M. Al-Murad, "Compressive sensing based opportunistic protocol for exploiting multiuser diversity in wireless networks," *IEEE International Symposium on Personal, Indoor and Mobile Radio Comm.*, 2009.
- [35] S. T. Qaseem and T. Y. Al-Naffouri, M. E. Eltayeb, and H. R. Bahrami, "Compressive sensing for feedback reduction in MIMO broadcast channels," *submitted to IEEE Trans. Wireless Comm.*
- [36] S. R. Bhaskaran, L. Davis, A. Grant, S. Hanly, and P. Tune, "Downlink scheduling using compressed sensing," *IEEE Inf. Theory Workshop on Networking and Inf. Theory*, pp. 201-205, 2009.
- [37] I. E. Nesterov, A. Nemirovskii, and Y. Nesterov, *Interior-point polynomial algorithms in convex programming*. SIAM, 1994.
- [38] Ali H. Sayed, *Fundamentals of adaptive filtering*. John Wiley & Sons, Ltd., 2003.

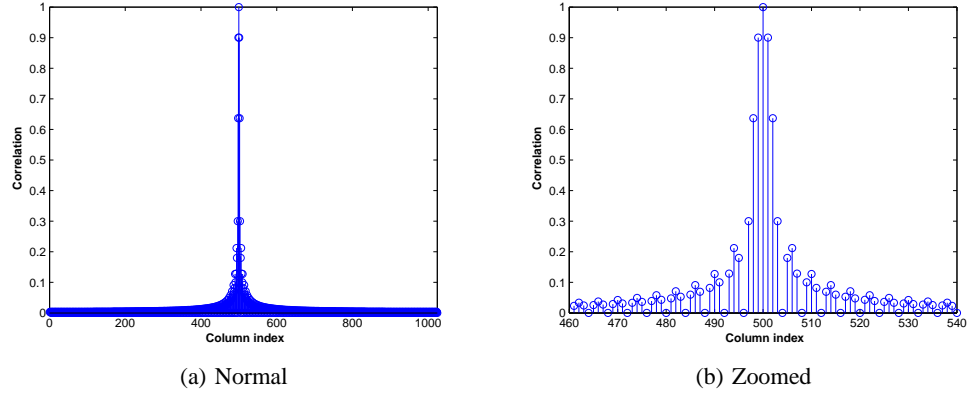


Fig. 1: The 500<sup>th</sup> column has high correlation with its neighbors

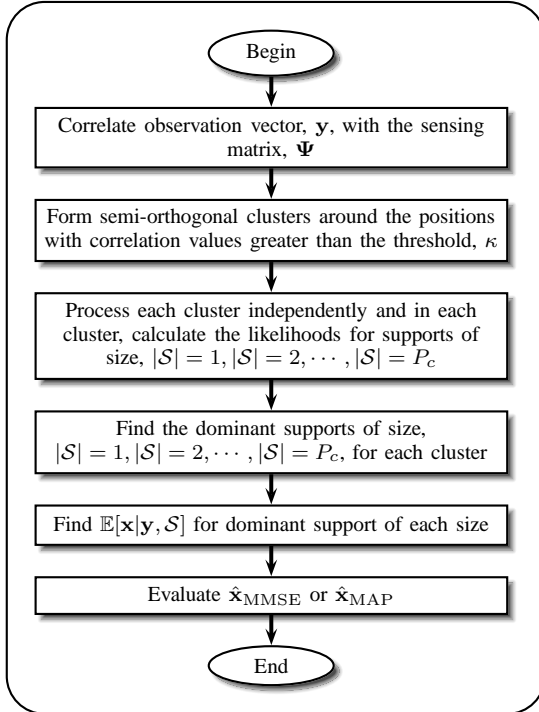


Fig. 2: Flowchart of the OC algorithm

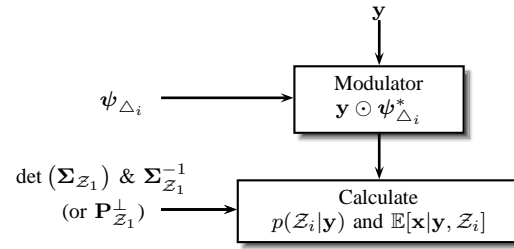


Fig. 3: Block diagram of the reduced complexity algorithm for the DFT matrix

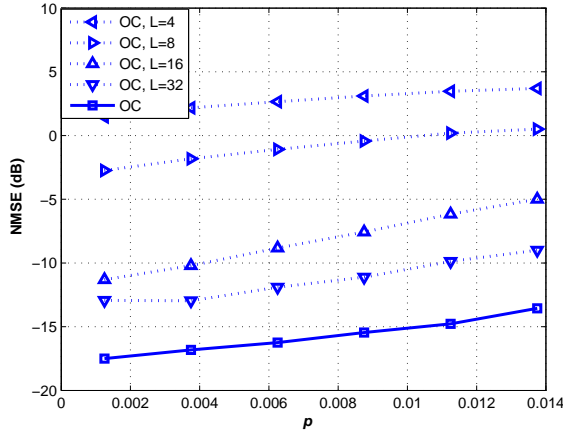


Fig. 4: NMSE vs  $p$  for the OC algorithm with the length of the cluster varied.

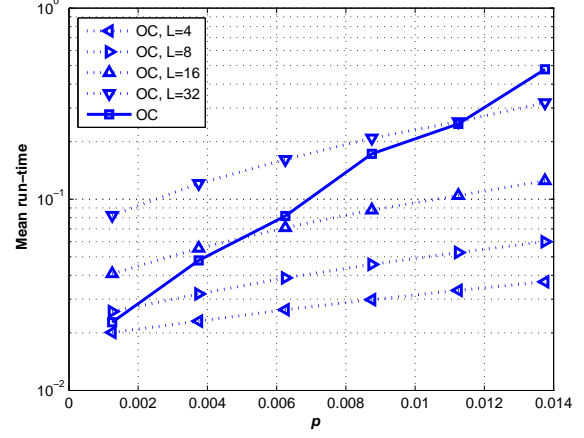


Fig. 5: Mean run-time for the OC algorithm with the length of cluster varied.

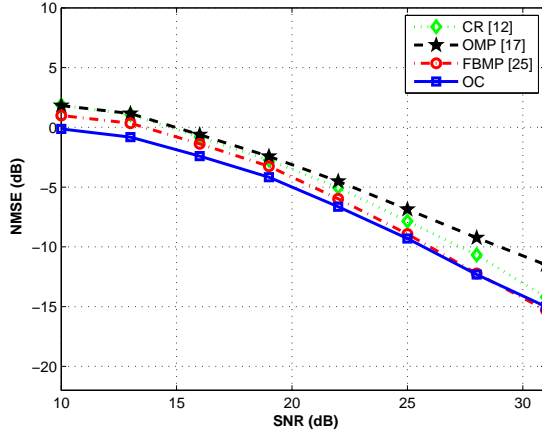


Fig. 6: NMSE vs SNR for the DFT matrix and  $\mathbf{x}|\mathcal{S}$  Gaussian.

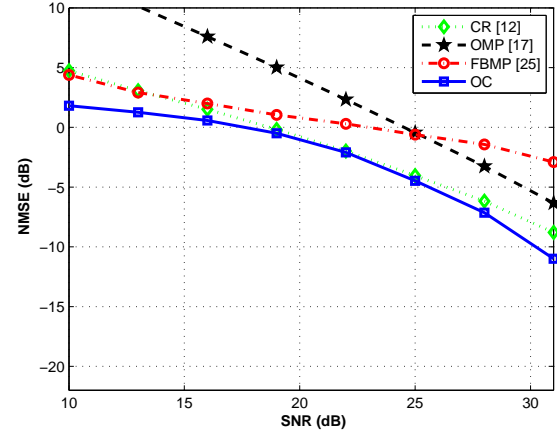


Fig. 7: NMSE vs SNR for the DFT matrix and  $\mathbf{x}|\mathcal{S}$  unknown.

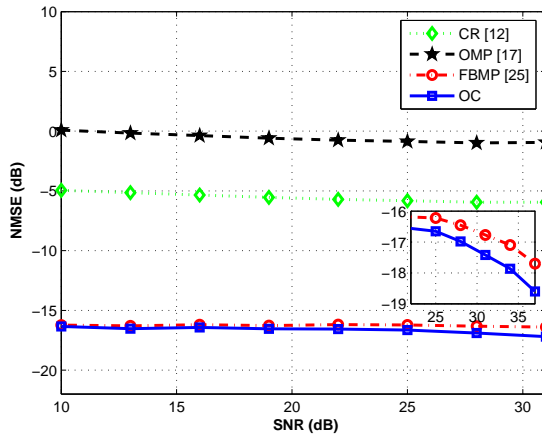


Fig. 8: NMSE vs SNR for the Toeplitz matrix and  $\mathbf{x}|\mathcal{S}$  Gaussian.

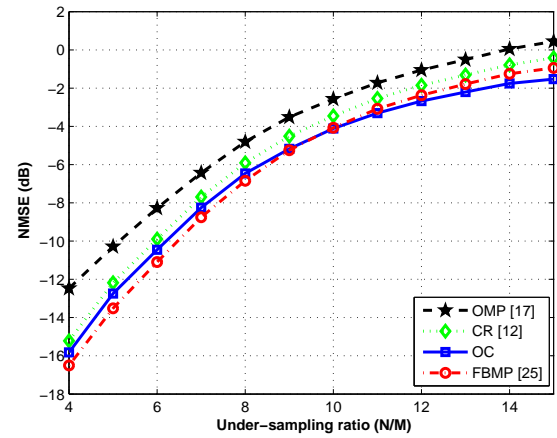


Fig. 9: NMSE vs the undersampling ratio ( $\frac{N}{M}$ ) for the DFT matrix and  $\mathbf{x}|\mathcal{S}$  Gaussian.

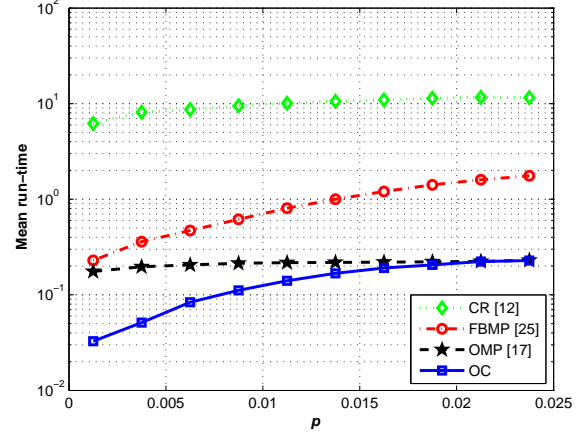
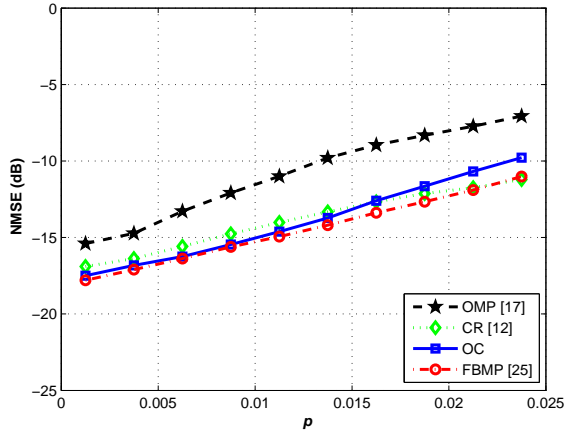


Fig. 10: NMSE vs  $p$  for the DFT matrix and  $\mathbf{x}|\mathcal{S}$  Gaussian. Fig. 11: Mean run-time for the DFT matrix and  $\mathbf{x}|\mathcal{S}$  Gaussian.

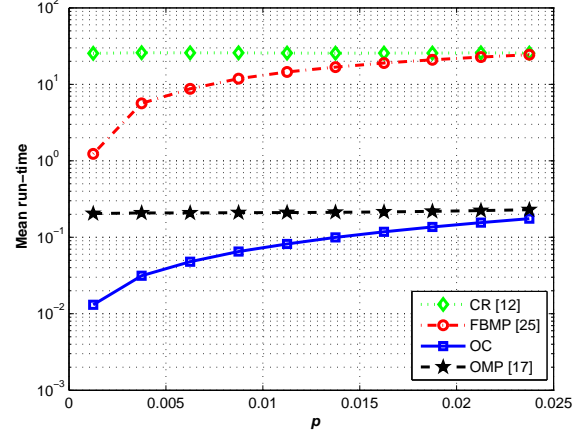
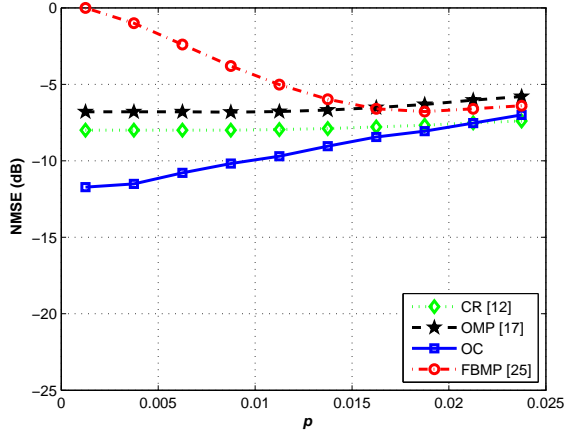


Fig. 12: NMSE vs  $p$  for the DFT matrix and  $\mathbf{x}|\mathcal{S}$  unknown. Fig. 13: Mean run-time for the DFT matrix and  $\mathbf{x}|\mathcal{S}$  unknown.

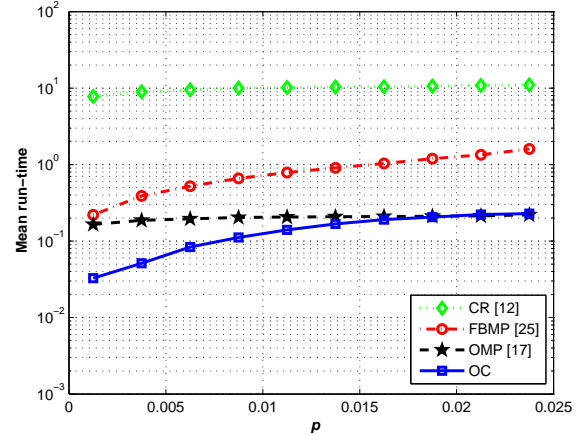
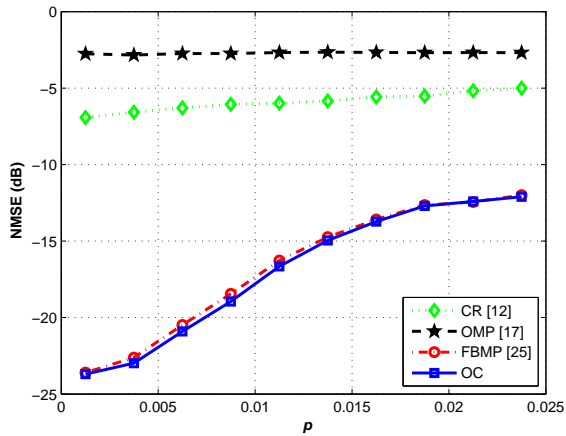


Fig. 14: NMSE vs  $p$  for the Toeplitz matrix and  $\mathbf{x}|\mathcal{S}$  Gaussian. Fig. 15: Mean run-time for the Toeplitz matrix and  $\mathbf{x}|\mathcal{S}$  Gaussian.